

AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions of claims in the application:

Listing of Claims:

1. (Currently Amended) A computer executable data structure comprising:
a computer processor coupled to a memory for executing the following,
a first data structure that describes one or more classes which define programmatic objects;
a second data structure that describes members of each class and comprises compound members that allow mapping of complex members as inline members of a given class, which allows inline mapping of arrays, structs and entity key members; and
a third data structure that describes relationships between objects, ~~wherein the described classes, members, and relationships provide information that can be utilized by a computer to~~
and persists ~~persist~~ object data to a database, wherein an object schema that describes data classes as well as relations between the data classes as specified in an object oriented model, is generated and utilized together with a relational schema and a mapping schema to map the programmatic objects to tables in the database;
wherein the mapping schema provides the mapping between the object schema and the relational schema, and the relational schema utilizes metadata associated with the database to generate an implementation neutral format or an implementation specific format that represents the database structure;
~~wherein members of a class include fields and properties; and~~
wherein the member properties include an alias attribute that is employed by a query language to identify a private member used to generate a query, the alias points to a public member that is to be utilized in place of the associated private member in text of a query; and
wherein the member properties include a hidden attribute that defines if there is a hidden member in a corresponding class and manages the hidden member in a transparent fashion.
2. (Canceled)

3. (Previously Presented) The data structure of claim 1, wherein a field includes a key attribute that defines whether the field is an object key.
4. (Previously Presented) The data structure of claim 1, wherein the properties include a path attribute that delimits the context of a class.
5. (Canceled)
6. (Original) The data structure of claim 1, wherein the members are compound members comprising members and other compound members.
7. (Original) The data structure of claim 6, wherein the compound member is an array.
8. (Original) The data structure of claim 6, wherein the compound member includes a type attribute that defines the type of data identified by the compound member.
9. (Original) The data structure of claim 1, wherein the third structure includes a type attribute that defines relationships between objects.
10. (Original) The data structure of claim 9, wherein the relationship is one of one-to-one, one-to-many, or many-to-many.
11. (Original) The data structure of claim 1, wherein the database is a relational database.
12. (Original) The data structure of claim 1, wherein the first, second and third data structures are XML structures.

13. (Currently Amended) An object schema generation system comprising:
a computer processor coupled to a memory for executing the following components,
a code reader component adapted to read or retrieve code from an object-oriented program or set of programs, the program describes objects *via* classes and class members;
an object schema generation component that retrieves or is provided with code from the code reader component, code is provided in real time as it is being read or transferred en masse upon a complete reading of the code, the object schema generation component produces an object schema in an extendible markup language (XML) which provides metadata concerning objects to facilitate persistence of object data to a data store, ~~such as a relational database,~~ wherein the generated object schema is utilized together with a relational schema and a mapping schema to map object data to tables in the data store;
wherein the mapping schema provides the mapping between the object schema and the relational schema, and the relational schema utilizes metadata associated with the data store to generate an implementation specific format that represents the data store structure; and
~~wherein the object schema provides information concerning classes, members of classes, and their relationships and~~ wherein properties of the members of classes include an alias attribute that is employed by a query language to identify a private member used to generate a query, the alias points to a public member that is to be utilized in place of the associated private member in text of a query; ~~and~~
wherein the object schema generation component utilizes rule based artificial intelligence to provide heuristics necessary to build the schema; and
wherein properties of the members of classes include a hidden attribute that defines if there is a hidden member in a corresponding class and manages the hidden member in a transparent fashion.
14. (Original) The system of claim 13, further comprising a data store information component adapted to provide the schema generation component with information concerning the data store.
15. (Canceled)

16. (Original) The system of claim 15, wherein the program is specified in an object oriented language.
17. (Original) The system of claim 16, wherein the program contains a plurality of object classes and fields.
18. (Canceled)
19. (Canceled)
20. (Canceled)
21. (Original) The system of claim 13, wherein the object schema generation component employs a Bayesian network to infer proper schema structures and relationships.
22. (Currently Amended) A method for producing an object schema comprising:
employing a processor coupled to a memory to execute the production of the object schema, comprising:
specifying classes to be persisted to a data store;
identifying members of each class, wherein the members of each class comprise compound members that allow mapping of complex members as inline members of a given class, which allows inline mapping of arrays, structs and entity key members;
specifying relationships between classes, ~~wherein the specified classes, members, and relationships provide information that can be utilized by a computer to persist and persisting~~
object data to the data store; ~~and wherein an object schema that describes data classes as well as relations between the data classes as specified in an object oriented model, is generated and utilized together with~~
utilizing an object schema, a relational schema and a mapping schema to map the object data to tables in the data store;
~~wherein the mapping schema provides the mapping between the object schema and the relational schema, and the relational schema utilizes~~ utilizing metadata associated with the

database to generate an implementation neutral or an implementation specific format that represents the data store structure; and

identifying a name of a member to be used as an alias to query a private member, ~~wherein an alias attribute that is employed by a query language to identify a private member is used to generate a query~~, the alias points to a public member that is to be utilized in place of the associated private member in text of a query; and

defining a hidden member in a corresponding class and managing the hidden member in a transparent fashion.

23. (Original) The method of claim 22, wherein the classes represent objects defined by an object oriented language.

24. (Original) The method of claim 23, wherein the data store is a relational database.

25. (Original) The method of claim 22, further comprising specifying a member key.

26. (Canceled)

27. (Original) The method of claim 22, wherein the member associated with a class is a field or property.

28. (Original) The method of claim 27, wherein the member is a compound member comprising at least one field or property and another compound member.

29. (Original) The method of claim 28, wherein the compound member is an array.

30. (Original) The method of claim 22, wherein the relationship between classes is one of one-to-one, one-to-many, and many-to-many.

31. (Original) The method of claim 22, wherein specifying class relationships comprises specifying a parent class and a child class.

32. (Original) The method of claim 31, further comprising specifying child members associated with the parent and child classes.

33. (Original) A computer readable medium having stored thereon computer executable instructions for carrying out the method of claim 22.

34. (Currently Amended) A method for generating an object schema comprising:
employing a processor coupled to a memory to execute the generation of the object schema, comprising:

receiving program code that describes one or more classes which define objects;
describing members of each class, wherein the members of each class comprise compound members that allow mapping of complex members as inline members of a given class, which allows inline mapping of arrays, structs and entity key members;

receiving input from a developer;
generating an object schema to be employed to facilitate mapping object components from an object oriented program to tables in a relational database, ~~wherein the generated object schema is utilized together with a relational schema and a mapping schema to map object components to the tables in the database;~~

providing a wherein the mapping schema that provides a the mapping between the object schema and a the relational schema, and the relational schema utilizes metadata associated with the database to generate an implementation neutral or an implementation specific format that represents the database structure;

~~wherein members of a class include fields and properties; and~~
identifying a name of a member to be used as wherein the member properties include an alias to query attribute that is employed by a query language to identify a private member used to generate a query, the alias points to a public member that is to be utilized in place of the associated private member in text of a query; and

defining a hidden member in a corresponding class and managing the hidden member in a transparent fashion.

35. (Original) The method of claim 34, wherein the developer provides input *via* a graphical user interface.
36. (Canceled)
37. (Original) The method of claim 34, wherein the schema is an XML schema.
38. (Original) The method of claim 34, wherein receiving input from a developer comprises identifying classes to be persisted and specifying relations amongst classes.
39. (Original) A computer readable medium having stored thereon computer executable instructions for carrying out the method of claim 34.